

Overriding method is being chosen at run time, but (in Java) already the compiler checks types

Overloading is also a form of polymorphism: the method body is determined during compilation

In a more specific sense, polymorphism is related to overriding: the method body is determined during execution

Whoever may operate upon the energy attribute. That's not good.

Interface:

- behavior abstraction
- behavior prescription

What needs to be restricted?

1. reading the attribute
2. writing the attribute
3. both reading and writing the attribute

Programming to an interface

A knight can have a sword.

Aggregation:

one object contains another

How to provide for this programmatically?

Aggregation generates hierarchy (has a):
an aggregate is superior to the parts

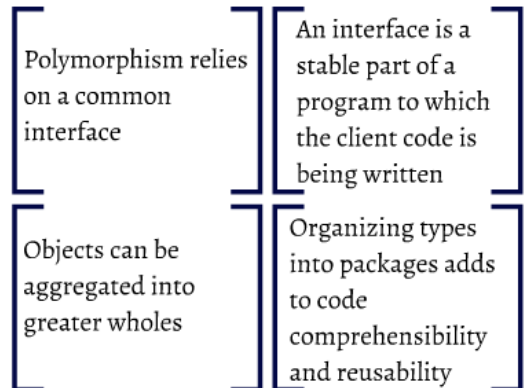
Inheritance also generates a hierarchy (is a):

a type (a class or interface) is superior to the derived types

If code has to be reused, what extent is the most appropriate?

1. a code block within a method
2. a method
3. a type (a class or interface)
4. several types

A package merges related types (classes and interfaces)



Lecture 2:

Polymorphism and Object-Oriented Modularization

Valentino Vranić

Ústav informatiky, informačných systémov a
softvérového inžinierstva



vranic@stuba.sk

fiit.sk/~vranic

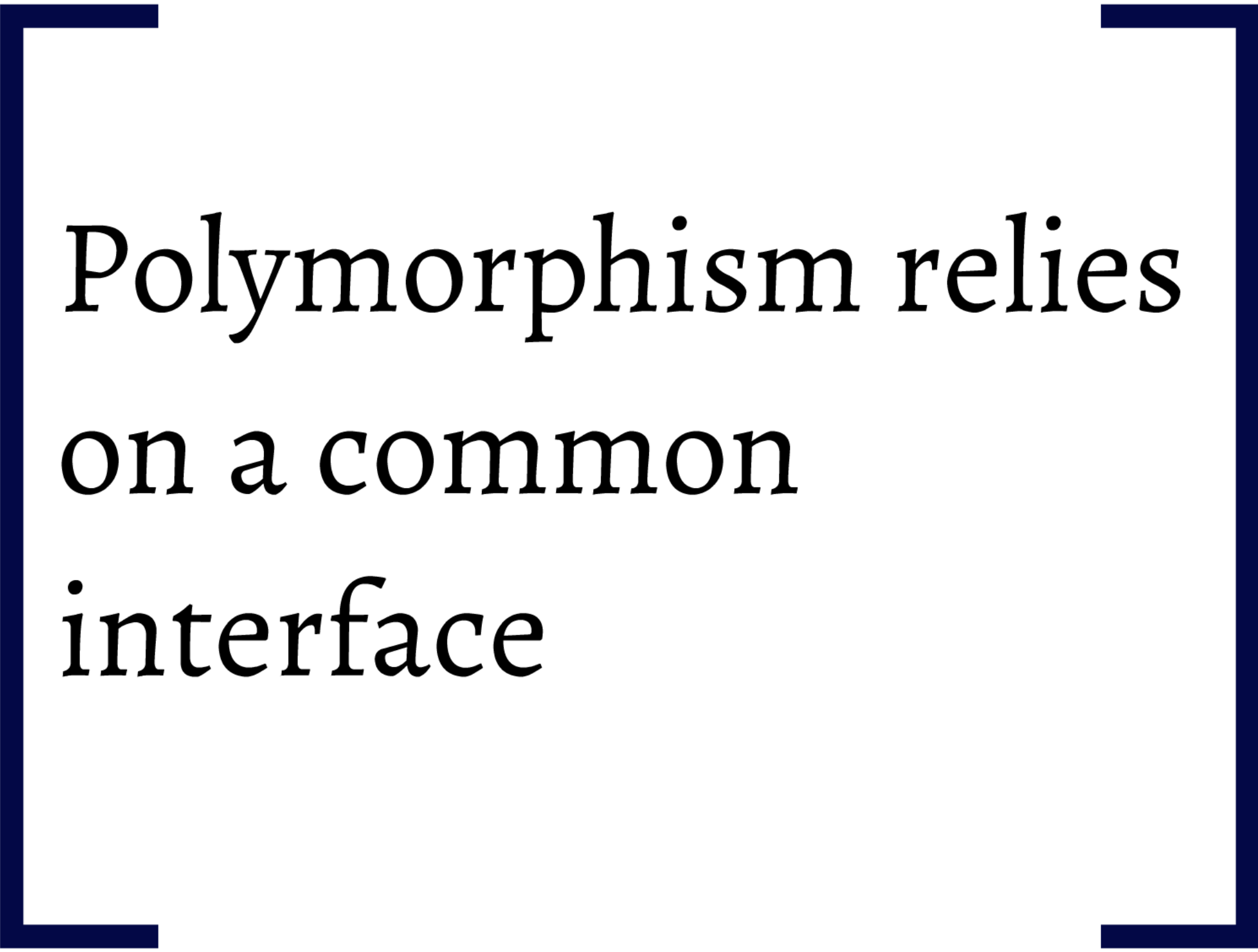
OOP 2018/19

19. 2. 2019

Overriding method is
being chosen at run time,
but (in Java) already the
compiler checks types

Overloading is also a form of polymorphism: the method body is determined during compilation

In a more specific sense, polymorphism is related to overriding: the method body is determined during execution



Polymorphism relies
on a common
interface

Whoever may operate upon the energy attribute. That's not good.

What needs to be restricted?

1. reading the attribute
2. writing the attribute
3. both reading and writing the attribute

Interface:

- behavior abstraction
- behavior prescription

Programming to an interface

An interface is a
stable part of a
program to which
the client code is
being written

A knight can have a sword.

How to provide for this
programmatically?

Aggregation:

one object contains another

Aggregation generates hierarchy (has a):

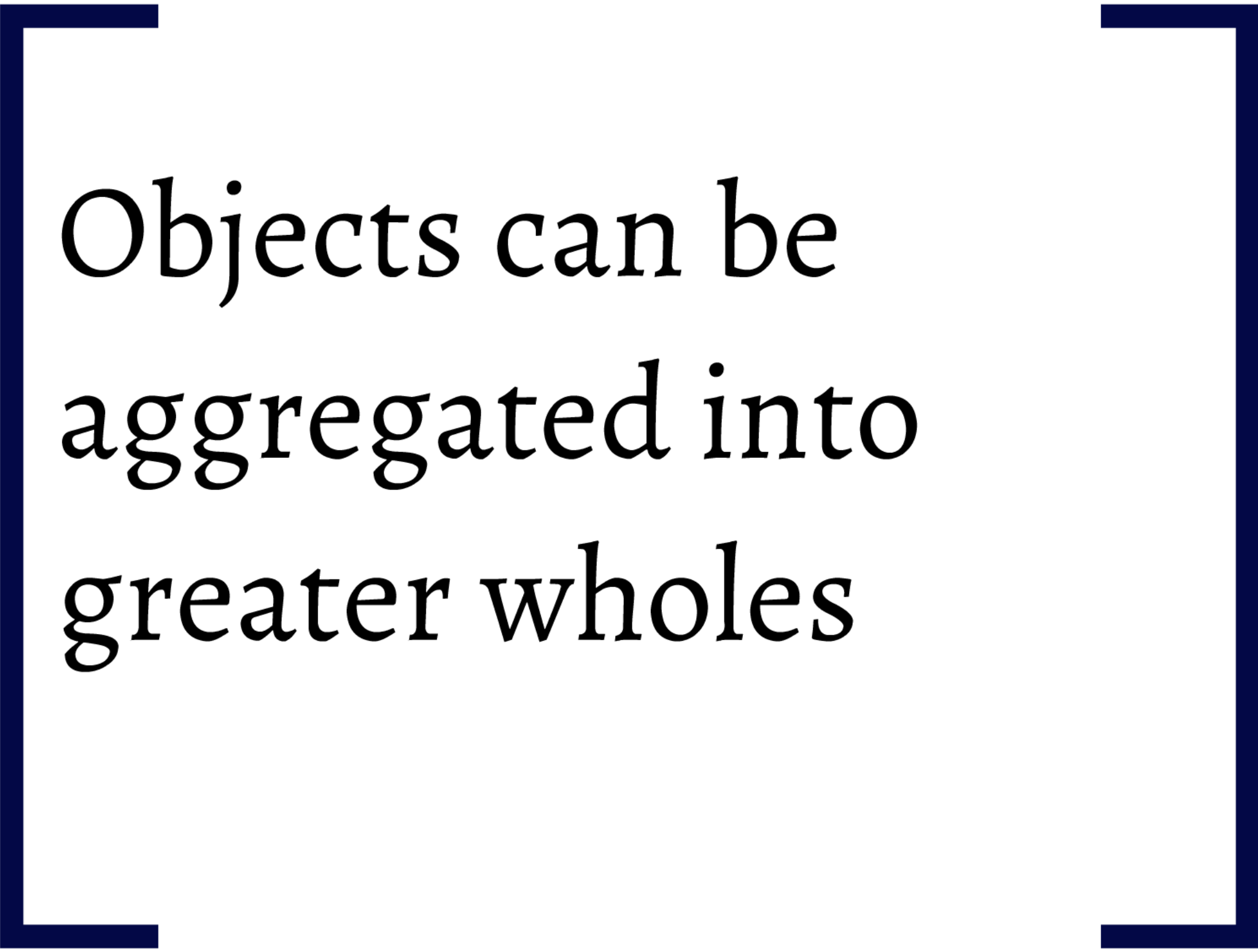
an aggregate is superior to the parts

Inheritance also generates a hierarchy

(is a):

a type (a class or interface) is

superior to the derived types



Objects can be
aggregated into
greater wholes

If code has to be reused, what extent is the most appropriate?

1. a code block within a method
2. a method
3. a type (a class or interface)
4. several types

A package merges related types (classes and interfaces)

Organizing types
into packages adds
to code
comprehensibility
and reusability

Polymorphism relies on a common interface

An interface is a stable part of a program to which the client code is being written

Objects can be aggregated into greater wholes

Organizing types into packages adds to code comprehensibility and reusability

Polymorphism relies
on a common
interface

An interface is a
stable part of a
program to which
the client code is
being written

Objects can be
aggregated into
greater wholes

Organizing types
into packages adds
to code
comprehensibility
and reusability